Comparison of Various Methods for the Calculation of the Distance Potential Field

Tobias Kretz, Cornelia Bönisch, and Peter Vortisch

PTV AG
Stumpfstraße 1
D-76131 Karlsruhe, Germany

{Tobias.Kretz,Cornelia.Boenisch,Peter.Vortisch}@PTV.De

April 24, 2008

Abstract

The distance from a given position toward one or more destinations, exits, and way points is a more or less important input variable in most models of pedestrian dynamics. Except for the special case when there are no obstacles in a concave scenario – i.e. each position is visible from any other – the calculation of these distances is a non-trivial task. This isn't that big a problem, as long as the model only demands the distances to be stored in a *Static Floor Field* also called *Potential Field* [1, 2, 3, 4, 5, 6, 7, 8], which never changes throughout the whole simulation. In this case a pre-calculation once before the simulation starts is sufficient. But if one wants to allow changes of the geometry during a simulation run – imagine doors or the blocking of a corridor due to some hazard – in the *Distance Potential Field*, calculation time matters strongly. This contribution gives an overview over existing and new exact and approximate methods to calculate a potential field, analytical investigations for their exactness, and tests of their computation speed. The advantages and drawbacks of the methods are discussed.

1 Introduction

1.1 General Introduction

The will to move through space is the will to reach some kind of destination. On a smaller time scale this is the will to reduce the distance toward some kind of destination. Therefore it appears to be natural to use the gradient of

the distance toward a destination in some kind of measure as primary input and impetus for motion in the simulation of the movement of pedestrians.

It is assumed that it is always sufficient to have a discrete distance potential field, either because the model itself is formulated in a discrete manner, or because some finite – yet arbitrarily large – exactness of the distance potential field is sufficient. Although potential fields are by no means confined to rectangular grids [9], only rectangular grids are investigated.

1.2 Robots and Pedestrians

Note that the potential discussed in this contribution has the meaning and is used in the way as in the simulation of pedestrian dynamics [3, 4, 5, 6, 7, 8, 10] and not robotics [1, 2]. The difference is that in robotics it is usually assumed that an *autonomous robot* knows about his destination coordinate but has no knowledge of the position of obstacles except for those which it "sees". For pedestrians on the contrary it is typically assumed that they have at least some knowledge on the whole path, the positions of obstacles and the detours they have to walk compared to linear distance, even if they actually only see a small fraction of the whole path. This has consequences for the calculation and use of the potential.

In robotics the *artificial potential* at position $\vec{x}$ from a destination at $\vec{x}_d$ was originally [1] calculated as

$$U_{artificial}(\vec{x}) = U_{\vec{x}_d}(\vec{x}) + U_{obstacles}(\vec{x}) \quad (1)$$

$$U_{\vec{x}_d}(\vec{x}) = \frac{1}{2} k_d (\vec{x} - \vec{x}_d)^2 \quad (2)$$

leading to a potential as shown in figure 1. In such a potential local minima can occur and a robot has to be equipped with the ability to realize it is in a minimum and how to get out of it.

Figure 1: Example for $U_{\vec{x}_d}(\vec{x})$ from equation (2).

Much different from this is the basic assumption in many models of pedestrian motion that pedestrians have a good global knowledge of the exact Euclidean distances and know about the shortest path between their current position and their destination under consideration of all obstacles.

This assumption can be unrealistic for very complex geometries like huge mazes (compare figure 2), but for many situations it comes close to reality.

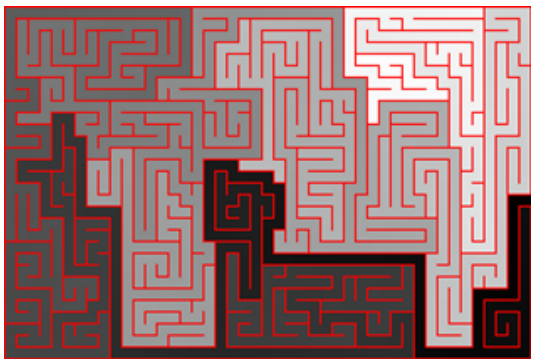


Figure 2: For pedestrians the *static floorfield* or *distance potential* is calculated differently to account for the global knowledge pedestrians have in most situations. Mazes pose an exception inasmuch as the essence of their existence is that it is hard, if not impossible, to acquire perfect global knowledge.

Note: the notion *Distance Potential Field* was chosen to clarify that it is nothing but a direct look-up table for the distance, whereas the *Static Floorfield* is inversely proportional to the distance and therefore – although edited in a trivial way – a quantity that's derived from distance to destination.

2 Methods for the Calculation of a Distance Potential Field

Short Mathematical Parenthesis: Vector Norms

In two dimensions the so-called *p-norms* are defined as

$$||\vec{x}||_p := (|\Delta x|^p + |\Delta y|^p)^{\frac{1}{p}} \text{ , with } p \in \mathbb{R}_+ \tag{3}$$

of these $p = 2$ is the well known *Euclidean Metric* (Pythagorean Theorem). So, for $p = 2$ the norm has the everyday life meaning of the word "distance". At first two other well known norms, which are relevant here, will be discussed: the ones with $p = 1$ and the limit $p \to \infty$. In principle the problem can be stated this way: only the metrics for $p = 1$ and $p \to \infty$ can be calculated by a flood fill, what one wants, however, is the metric with $p = 2$ for which no equally fast and simple calculation method exists.

In this paper the following notation is used: the distance D from a certain point to the exit on the shortest path for a method X is called D^X. It is composed of a sum of distances d_i^X of (straight) line segments of the visibility graph [11], i.e. they connect sequently the exit, corners of obstacles, and

the coordinate under consideration. Note that for the flood fill methods the visibility graph does not have to be known explicitly for the calculations.

2.1 Flood Fill Methods

In *Flood Fill* (sometimes also called *Wavefront*) methods the distance is calculated by sequently moving cell to closest neighbor cell and by that summing up the distances.

2.1.1 Manhattan Metric

For $p = 1$ one has the famous *Manhattan Metric* – also called *Taxicab Metric* or *Manhattan Distance*. It was introduced by Hermann Minkowski – but note that the name "Minkowski Metric" is reserved for the elementary metric of special relativity. Just as a taxi driver in Manhattan needs to sum up the number of Streets and the number of Avenues which he has to cross during the drive to get an estimation of the distance, the distance D^M according to the Manhattan metric simply is the sum of the absolute values of the differences in x- and y-coordinate. Flood Fill therefore only acts within the *von Neumann Neighborhood*, i.e. grid cells need to be connected via a common edge (see figure 3(a)).

$$\Delta x = \sum_i |\delta x_i| \text{ and } \Delta y = \sum_i |\delta y_i| \tag{4}$$

$$D^M = \sum_i d_i^M = \Delta x + \Delta y \tag{5}$$

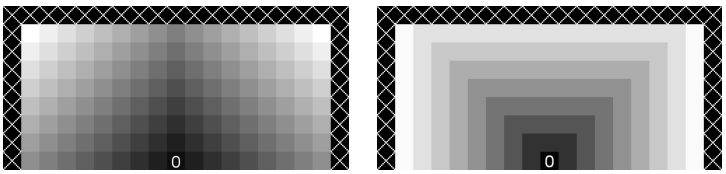
2.1.2 Chessboard Metric

For the limit $p \to \infty$ one arrives at the *Maximum Norm* - also called *Chebyshev Distance* D^C or *Chessboard Metric*. It was introduced by Pafnuty Chebyshev and got its alternative names from the way the king in chess is allowed to move, respectively the fact that the distance measured with this norm is the maximum of the differences in x- or y-coordinate. In other words: flood fill acts within the *Moore Neighborhood* and the added value is the same (typically 1) whether grid cells are connected via an edge or a corner (see figure 3(b)).

$$d_i^C = \max\left(|\delta x_i|, |\delta y_i|\right) \tag{6}$$

$$D^C = \sum_i d_i^C \tag{7}$$

The advantage of Manhattan and Chessboard metric is the fact that the typically very fast flood fill procedure can be applied. For $2 < p < \infty$ it is in general not possible to calculate the correct distance (i.e. the Euclidean



(a) Manhattan metric (b) Chessboard metric

Figure 3: Manhattan and Chessboard metric.

distance) between two arbitrary grid cells with a flood fill with local rules. The major drawback of course is that aside from Manhattan, Mannheim, or chess there are only few occasions where one of the two norms gives an exact result. However, there are possibilities to stick with the flood fill method and gain exactness. Some of these will be discussed in the following.

2.1.3 Variant 1: Combination of Manhattan and Chessboard

There is a method which uses flood fill and which is exact in the Euclidean sense for all positions which are visible from the destination. Since Manhattan metric gives $d^M_{i=1} = |\delta x_{i=1}| + |\delta y_{i=1}|$ and Chessboard metric results in $d^C_{i=1} = \max(|\delta x_{i=1}|, |\delta y_{i=1}|)$, one can simply calculate $d^m_{i=1} = d^M_{i=1} - d^C_{i=1} = \min(|\delta x_{i=1}|, |\delta y_{i=1}|)$. This "Minimum Norm" (see figure 4(a)) is of no use in itself, but $(d^m_{i=1})^2 + \left(d^C_{i=1}\right)^2$ must equal $(\delta x_{i=1})^2 + (\delta y_{i=1})^2$, as one of the two δs as trivially as necessarily needs to be the maximum and the other one the minimum. And $(\delta x_{i=1})^2 + (\delta y_{i=1})^2$ is the square of the exact Euclidean distance ($p = 2$) between the exit and a coordinate which is directly visible from the exit. If one generalizes this for line segments which are not directly visible from the exit ($i > 1$) one makes an error, since one can only calculate the square root of the sum of squared line segment (Euclidean) distances, while the exact result would be the sum of the square roots of squared line segment (Euclidean) distances:

$$\sqrt{\sum_i (d^E_i)^2} \neq \sum_i \sqrt{(d^E_i)^2} \tag{8}$$

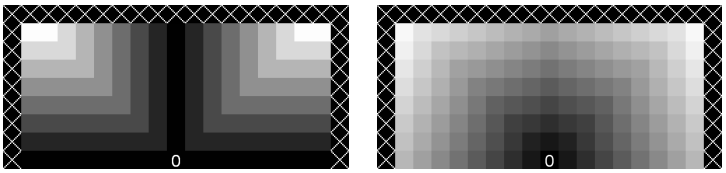
Regardless of this problem, equations (9) – (11) can be motivated by the initial observation that the calculation of distances is exact "to the point of the next corner". This is achieved at the expense of making two flood fills – the final value of D^{V_1} can not be calculated by a single flood fill – and

having to calculate a square root for each cell in the end.

$$d_i^m = d_i^M - d_i^C = \min\left(|\delta x_i|, |\delta y_i|\right) \tag{9}$$

$$D^m = \sum_i d_i^m = D^M - D^C \tag{10}$$

$$\left(D^{V_1}\right)^2 = \left(D^C\right)^2 + (D^m)^2 \tag{11}$$



(a) "Minimum metric" (b) Variant 2

Figure 4: "Minimum metric" and metric for variant 2

2.1.4 Variant 2: $\sqrt{2}$ over Corners

Another simple modification is a Chessboard flood fill where flooding across corners adds a $\sqrt{2}$ instead of 1. If one does so, not only distances parallel to the discretization axis will be exact, but also distances deviating by exactly 45° from that. A version of such a modification, which additionally includes a smoothing mechanism, is introduced in [12]. Since the number of diagonal steps is $\min(\delta x_i, \delta y_i)$ and the number of horizontal or vertical steps is $\max(\delta x_i, \delta y_i) - \min(\delta x_i, \delta y_i)$ one can write

$$d_i^{V_2} = \sqrt{2} d_i^m + \left(d_i^C - d_i^m\right) \tag{12}$$

$$D^{V_2} = \sum_i d_i^{V_2} = \left(\sqrt{2} - 1\right) D^m + D^C \tag{13}$$

2.1.5 Variant 3: Larger Neighborhoods

A modification which gains computation speed on the cost of exactness is to increase the neighborhood further, so cells with a distance of two or three or even more are included. As one only has to continue the calculation with the border cells as new center cells, the recursion depth or stack size is reduced, but one runs the risk of overlooking small obstacles. This method is not investigated further here.

2.2 Dijkstra's Algorithm on a Visibility Graph

Another method is to try to find a subset of grid cells which form a *Visibility Graph* [7, 11, 13] (compare figure 5). These grid cells are all grid cells which are necessary as navigation cell for at least one arbitrary grid cell if a

pedestrian wants to move from that arbitrary grid cell around the obstacles to the destination. Two nodes of the visibility graph are connected if and only if they are mutually visible. Once one has created such a visibility graph, one can calculate the distance toward the destination for all grid cells which are part of the visibility graph using *Dijkstra's Algorithm* [14]. After having done that one can calculate the distance from all other grid cells toward the destination by making use of the visibility graph and the distance information now contained within it. Note that strictly spoken the method used to measure the computation time is the one from [13] and not from [7, 11], where the latter one is probably more efficient.

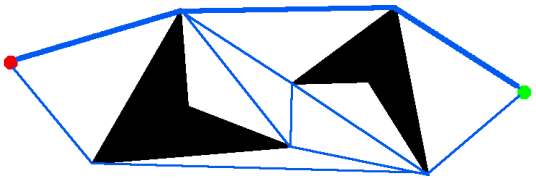


Figure 5: Visibility graph of a simple geometry.

2.3 Ray Casting

Another very intuitive method is an iterated *Ray Casting*. In detail, the following algorithm is applied:

1. Calculate the distance toward the destination for all grid cells which are visible from the destination.

2. If possible, find that grid cell X_0, which is the closest to the destination of all those grid cells which have been assigned a distance, but which have at least one neighbor which is neither an obstacle, nor has been assigned a distance toward the destination, or has been assigned a distance, but a distance which is too large.

3. Calculate the distance toward the destination of all cells which are visible from X_0. If there are cells which are visible to X_0 and which have already been assigned a distance toward the destination, this distance is only overwritten, if the newly calculated distance is smaller.

4. Repeat steps 2. and 3. until step 2. does not find any grid cell anymore.

It is important for the calculation time how the check for visibility is done. Concerning the ray tracing part one can use the *Bresenham Line Drawing Algorithm* [15]. But it is important that this algorithm is not used to draw a line (cast a ray) each time one wants to calculate the visibility of a grid cell. It is better to draw a rectangle around the whole scenario

("border") and cast rays from the cell in focus to each of those border cells. All cells before the first obstacle are then marked as visible, all behind as "not visible". Because of the discreteness, it might occur that the ray casting toward neighbored border cells gives different results for the visibility of some cell (when a cell is part of multiple casts). In that case, the cell needs to be marked as visible, because otherwise in scenarios with narrow spaces "blind spots" can appear, of which the algorithm would claim that they are not accessible at all. With this strategy the number of lines, one has to draw, only grows as the border size instead of as the area.

2.4 Other Methods of Error Reduction

One has to distinguish the *distance error* in the distance potential field from the *speed error* in a model. If the pedestrians move on a discrete lattice as well, the lattice structure can lead to errors in the speed, just as it leads to errors in the distance. The speed errors can be compensated for by making pedestrians suspend certain moves depending on the ratio of corner versus edge steps they did in recent moves. Such strategies are proposed for example in [12, 16].

3 Analytical Considerations

3.1 Errors for Manhattan and Chessboard Metric

The maximal errors of the two simple metrics are both: well known and trivial to calculate, nevertheless for the sake of completeness, they are given in the following. The absolute error compared to the Euclidean distance d_i^E of a single straight line element of some path using the Manhattan metric is

$$\begin{aligned} e_i^M &= d_i^M - d_i^E && (14)\\ &= d_i^E(|\cos\varphi_i| + |\sin\varphi_i| - 1) && (15)\\ \text{with } d_i^E &= \sqrt{|\delta x_i|^2 + |\delta y_i|^2} && (16) \end{aligned}$$

with φ_i being the angle between the connecting line of the two points and the x-axis. This simply sums up to a total error of

$$E^M = \sum_i d_i^E(|\cos\varphi_i| + |\sin\varphi_i| - 1) \qquad (17)$$

which always lies between the boundaries

$$\begin{aligned} 0 &\leq E^M \leq (\sqrt{2}-1)D^E && (18)\\ \text{with } D^E &= \sum_i d_i^E \qquad \text{(exact total Euclidean distance)} && (19) \end{aligned}$$

The maximal error arises from diagonal motion.

The corresponding values for the Chessboard metric are

$$\begin{aligned} e_i^C &= d_i^C - d_i^E && (20)\\ &= d_i^E(\max(|\cos\varphi_i|, |\sin\varphi_i|) - 1) && (21)\\ E^C &= \sum_i d_i^E(\max(|\cos\varphi_i|, |\sin\varphi_i|) - 1) && (22) \end{aligned}$$

with the latter one always satisfying the relation

$$(\sqrt{0.5} - 1)D^E \leq E^C \leq 0 \quad (23)$$

Here as well the extreme value is reached for diagonal motion.

3.2 Error for Variant 1 (Combination)

To calculate the (extremal) errors, one has to deal with in variant 1, is a bit more complicated than it is for the two basic metrics above. The error is

$$\begin{aligned} E^{V_1} &= D^{V_1} - D^E && (24)\\ &= \sqrt{\Big(\sum_i \max(|\delta x_i|, |\delta y_i|)\Big)^2 + \Big(\sum_i \min(|\delta x_i|, |\delta y_i|)\Big)^2} && (25)\\ &\quad - \sum_i \sqrt{|\delta x_i|^2 + |\delta y_i|^2} \end{aligned}$$

Remember that

$$|\delta x_i| = d_i^E|\cos(\varphi_i)| \text{ and } |\delta y_i| = d_i^E|\sin(\varphi_i)| \quad (26)$$

and use as abbreviation

$$\begin{aligned} M_i &= \max(|\cos(\varphi_i)|, |\sin(\varphi_i)|) && (27)\\ m_i &= \min(|\cos(\varphi_i)|, |\sin(\varphi_i)|) && (28) \end{aligned}$$

in equation (25).

$$\begin{aligned} E^{V_1} &= \sqrt{\sum_i \sum_j d_i^E d_j^E (M_i M_j + m_i m_j)} - \sum_i d_i^E && (29)\\ &= \sqrt{\sum_i \left(d_i^E\right)^2 + \sum_i \sum_{j\neq i} d_i^E d_j^E (M_i M_j + m_i m_j)} - \sum_i d_i^E && (30)\\ &= \sqrt{\Big(\sum_i d_i^E\Big)^2 + \sum_i \sum_{j\neq i} d_i^E d_j^E (M_i M_j + m_i m_j - 1)} - \sum_i d_i^E && (31)\\ &= D^E\left(\sqrt{1 + \frac{\sum_i \sum_{j\neq i} d_i^E d_j^E (M_i M_j + m_i m_j - 1)}{\left(\sum_i d_i^E\right)^2}} - 1\right) && (32) \end{aligned}$$

With the relations

$$0 \leq \quad m_i \leq \sqrt{0.5} \leq M_i \quad \leq 1 \tag{33}$$

$$\sqrt{0.5} \leq \quad m_i m_j + M_i M_j \quad \leq 1 \tag{34}$$

one gets for the error

$$D^E \left(\sqrt{1 + \left(\sqrt{0.5} - 1\right) \frac{\sum_i \sum_{j \neq i} d_i^E d_j^E}{\left(\sum_i d_i^E\right)^2}} - 1 \right) \leq E^{V_1} \leq 0 \tag{35}$$

which is extremal for direction changes from horizontal/vertical to diagonal or vice versa. No error at all arises, when the new direction can be generated from the old by a reflection at one of the axis or diagonals. The left part will take the most extreme value, if all of the d_i^E are equal. With N as the number of single straight line elements one gets

$$D^E \left(\sqrt{1 + \left(\sqrt{0.5} - 1\right) \frac{N-1}{N}} - 1 \right) \leq E^{V_1} \leq 0 \tag{36}$$

This confirms the initial observation that there is no error for $N = 1$. However, the error in this variant depends on the number of line elements and therefore the number of obstacles in – respectively the complexity of – the scenario. In the limit $N \to \infty$ the error can in the worst case (denoted by the hat) be

$$\hat{E}^{V_1} = D^E \left(0.5^{0.25} - 1\right) \approx -0.159 D^E \tag{37}$$

3.3 Error for Variant 2 ($\sqrt{2}$ over Corners)

For variant 2 the error is

$$E^{V_2} = \left(\sqrt{2} D^m + D^C - D^m\right) - D^E \tag{38}$$

$$= \left(\left(\sqrt{2} - 1\right) D^m + D^C\right) - D^E \tag{39}$$

$$= \sum_i \left(\left(\sqrt{2} - 1\right) \min(|\delta x_i|, |\delta y_i|) + \max(|\delta x_i|, |\delta y_i|) - d_i^E\right) \tag{40}$$

with equations (27) and (28) this is

$$E^{V_2} = \sum_i d_i^E \left(\left(\sqrt{2} - 1\right) m_i + M_i - 1\right) \tag{41}$$

$$= \sum_i d_i^E \left(\left(\sqrt{2} - 1\right) m_i + \sqrt{1 - m_i^2} - 1\right) \tag{42}$$

bearing in mind that $0 \leq m_i \leq \sqrt{0.5}$ one finds the maximum of each summand at

$$\hat{m}_i = \frac{\sqrt{2 - \sqrt{2}}}{2} \tag{43}$$

which corresponds to an angle of exactly $\hat{\phi} = \pi/8 = 22.5°$ and any corresponding angle in the other seven octants. The maximum error then is

$$\hat{E}^{V_2} = \left(\sqrt{4-2\sqrt{2}}-1\right) D^E \tag{44}$$

$$\approx 0.082 D^E \tag{45}$$

which is slightly better than the maximum error $\hat{E}_2^{V_1} \approx 0.099$ for variant 1 for $N = 2$ With this method there can never be a negative error. If one wants the error to vanish on the average of all angles, one can add the values

$$s_{hv} = \frac{\hat{\varphi}}{\alpha} = \frac{\pi}{8\left(\sqrt{2}-1\right)} \approx 0.948 \tag{46}$$

$$s_d = \sqrt{2}\frac{\hat{\varphi}}{\alpha} = \sqrt{2}\frac{\pi}{8\left(\sqrt{2}-1\right)} \approx 1.341 \tag{47}$$

when flooding horizontally or vertically respectively diagonally. I.e. there is a global factor of ≈ 0.948 multiplied to any distance, as $s_d/s_{hv} = \sqrt{2}$. In this case the directions $\varphi_{1,2}$ (of the first octant) with exact measurements are at

$$\sin\varphi_{1,2} = \frac{4-2\sqrt{2}}{\pi} \pm \sqrt{\frac{1}{4-2\sqrt{2}} - \frac{8}{\pi^2}} \tag{48}$$

which is approximately $\varphi_1 \approx 9.55°$ and $\varphi_2 \approx 35.45°$. In the range between these two angles distances are measured too large, outside of this range too small.

4 Computation Times

4.1 Geometries

The test geometries were – as shown in figure 6 – a "typical" room, a maze with 200 x 200 grid cells, a circle shaped room with a diameter of 996 grid cells , a square shaped room with a side length of 3998 grid cells, a room with a large column in the middle, and a ring, the latter two with the same size as the circle.

4.2 Results

The following tables give the results for computation time (standard PC) and the largest distance from the exit that was found by the method. Among the flood fill based methods variant 2 is different from all other flood fill methods regarding the fact that taking a square root for each grid cell is a necessary part of the calculation. For square roots and other elements of the calculation the calculation time not only depends on the algorithm but also on the details of the implementation.

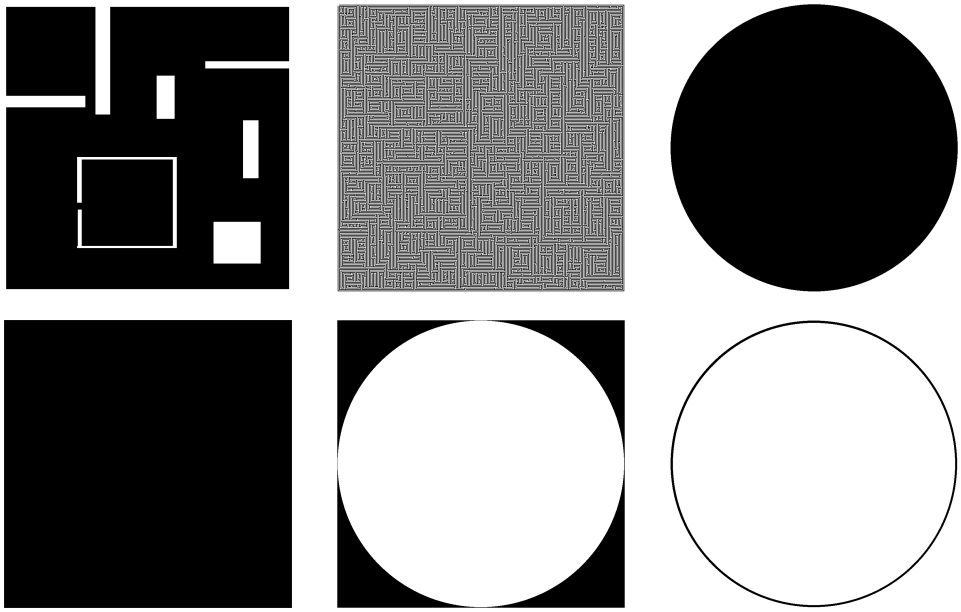


Figure 6: Test geometries. Walkable areas are colored black, walls white. Note that the scenarios were largely different in size and just scaled differently to fit the page (compare description in the text).

In the visibility graph method, the visibilities were calculated via the area method. This led to the comparatively large calculation times. Though most of the time they were smaller than those of the ray casting method with visibility calculation by the area method. The only exception was the circle scenario. The reason for this is that the node generating method presented in [13] generates many unnecessary nodes, as the bending of the circle's border was so small that many local neighborhoods might also have been part of a convex corner, although the inner border of a circle is entirely concave. If the circle had been given as geometric object this could have been avoided.

The ray casting method which uses the border method for visibility calculations was pleasingly fast, although never as fast as the Manhattan flood fill, and probably not yet fast enough to be applied each time step for many destinations.

5 Conclusions

In this contribution various methods for the calculation of distances in an obstacle filled space were investigated for their deviation from the true Euclidean distance and the time consumption for their calculation. Starting with the two well-known metrics – Manhattan and Chessboard – variants

Geometry: Typical Method	Calc. Time [s]	Maximal Distance	**Maze**	Calc. Time [s]	Maximal Distance
Manhattan	0.04	1246.00		0.00	7507.00
Chessboard	0.06	960.00		0.01	7198.00
Variant 1 (Comb.)	0.12	1001.70		0.02	7204.63
Variant 2 ($\sqrt{2}$)	0.08	1091.35		0.01	7325.97
Ray Casting (edge)	1.18	1052.85		72.72	7361.20
Visibility Graph	11.00	1052.16		386.2	7325.97
Ray Casting (area)	51.00	1053.72		1156	7507.00
Geometry: Circle			**Square**		
Manhattan	0.06	1202.00		2.44	7994.00
Chessboard	0.10	996.00		2.69	3997.00
Variant 1 (Comb.)	0.18	996.24		5.52	5652.61
Variant 2 ($\sqrt{2}$)	0.10	1037.52		2.68	5652.55
Ray Casting (edge)	0.34	996.24		4.29	5652.61
Visibility Graph	100.80	996.24		628.00	5652.61
Ray Casting (area)	19.00	996.24		1606.00	5652.61
Geometry: Column			**Ring**		
Manhattan	0.02	1993.00		0.01	1971.00
Chessboard	0.03	1408.00		0.01	1386.00
Variant 1 (Comb.)	0.06	1524.69		0.02	1504.40
Variant 2 ($\sqrt{2}$)	0.03	1650.30		0.01	1628.30
Ray Casting (edge)	8.89	1565.08		7.36	1542.24
Visibility Graph	47.80	1564.94		7.00	1542.09
Ray Casting (area)	163.00	1565.26		22.00	1542.39

Table 1: Computation time and maximal distance for the geometries of figure 6.

of such flood fill methods were reviewed respectively introduced. Following that a visibility graph and a ray casting method were discussed. It was found that compared to the standard metrics it is possible to significantly reduce the error while sticking to the flood fill method by making subtle changes to or using combinations of the standard metrics. The errors that remain balance with the fact that their calculation is significantly quicker than the calculation of the two other methods which are – in principle – error-free.

Acknowledgments:

For valuable discussion and hints we thank U. Hanebeck, S. Hengst, A. Pohlmann, and L. Spannehl.

References

[1] O. Khatib. The Potential Field Approach and Operational Space Formulation in Robot Control. In K.S. Narendra, editor, *Adaptive and Learning Systems – Theory and Application*, pages 367–377, New York and London, 1986. Plenum Press. urn:ISBN:978-0306422638.

[2] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 7th edition, 1991. urn:ISBN:978-0792391296.

[3] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittarz. Simulation of Pedestrian Dynamics Using a 2-dimensional Cellular Automaton. *Physica A*, 295:507, 2001. DOI:10.1016/S0378-4371(01)00141-8. arXiv:cond-mat/0102397v1.

[4] A. Schadschneider. Bionics-Inspired Cellular Automaton Model for Pedestrian Dynamics. In M. Fukui, Y. Sugiyama, M. Schreckenberg, and D.E. Wolf, editors, *Traffic and Granular Flow '01*, pages 499–509, Nagoya, 2003. Springer-Verlag Berlin Heidelberg.

[5] A. Kirchner and A. Schadschneider. Simulation of Evacuation Processes Using a Bionics-inspired Cellular Automaton Model for Pedestrian Dynamics. *Physica A*, 312(1-2):260–276, 2002. DOI:10.1016/S0378-4371(02)00857-9. arXiv:cond-mat/0203461v1.

[6] A. Kirchner. *Modellierung und statistische Physik biologischer und sozialer Systeme*. PhD thesis, Universität zu Köln, 2002. (in German). urn:nbn:de:hbz:38-112313296.

[7] K. Nishinari, A. Kirchner, A. Namazi, and A. Schadschneider. Extended Floor Field CA Model for Evacuation Dynamics. *IEICE Trans. Inf. & Syst.*, E87-D:726–732, 2004. arXiv:cond-mat/0306262v1.

[8] T. Kretz and M. Schreckenberg. The F.A.S.T.-Model. In S. El Yacoubi, B. Chopard, and S. Bandini, editors, *Cellular Automata – 7th International Conference on Cellular Automata for Research and Industry, ACRI 2006, Proceedings*, pages 712–715, Perpignan, France, September 2006. Springer-Verlag Berlin Heidelberg. DOI:10.1007/11861201_85. arXiv:0804.1893v1. urn:ISBN:978-3-540-40929-8.

[9] J. Meister. Simulation of crowd dynamics with special focus on building evacuations. Master's thesis, Fachhochschule Wedel, 2007. http://www.quovadis-simulation.de/.

[10] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation Dynamics: Empirical Results, Modeling and Applications. In R.A. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, Berlin Heidelberg New York, 2009. (to be published). urn:ISBN:978-0-387-75888-6. arXiv:0802.1620v1. http://refworks.springer.com/mrw/index.php?id=259.

[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, Berlin Heidelberg New York, 1997. urn:ISBN:3-54061-270-X.

[12] H. Klüpfel. *A Cellular Automaton Model for Crowd Movement and Egress Simulation*. PhD thesis, Universität Duisburg-Essen, 2003. urn:nbn:de:hbz:464-duett-08012003-0925403.

[13] T. Kretz. *Pedestrian Traffic - Simulation and Experiments*. PhD thesis, Universität Duisburg-Essen, 2007. urn:nbn:de:hbz:464-20070302-120944-7.

[14] E.W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959. DOI:10.1007/BF01386390.

[15] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4:25–30, 1965. http://www.research.ibm.com/journal/sj/041/ibmsjIVRIC.pdf.

[16] M. Schultz, S. Lehmann, and H. Fricks. A discrete microscopic model for pedestrian dynamics to manage emergency situations in airport terminals. In N. Waldau, P. Gattermann, H. Knoflacher, and M. Schreckenberg, editors, *Pedestrian and Evacuation Dynamics 2005*, pages 369–375, Vienna, 2006. Springer-Verlag Berlin Heidelberg. urn:ISBN:3-540-42690-6.